Remarks

Applicant respectfully requests that this Response After Final Action be admitted under 37 C.F.R. § 1.116.

Applicant submits that this Response presents claims in better form for consideration on appeal. Furthermore, applicant believes that consideration of this Response could lead to favorable action that would remove one or more issues for appeal.

No claims have been amended. No claims have been canceled. Therefore, claims 1-3, 6-14 and 16-27 are now presented for examination.

Claims 1-3, 7, 9, 10-14, 16-20, and 22-24 stand rejected under 35 U.S.C. §102(e) as being anticipated by Horvitz (U.S. Patent No. 6,009,452). Further, claim 21 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Horvitz. Applicant submits that the present claims are patentable over Horvitz.

Horvitz discloses that a task instance is selected at a beginning of each of idle-time intervals and processed during the remainder of that interval. The selected instance is one that is most likely to occur in the near-future and will be precomputed, during an associated idle-time interval, to the extent of the remaining time available during that interval. Once this task instance has been precomputed, its results, partial if the task has not completed prior to the end of the interval, are stored for subsequent access and use. Ideally, by having these results precomputed and ready for future use, future response time, i.e. run-time delay, is appreciably shortened since that task instance, when it would otherwise be dispatched for execution, will have already executed--fully or at least partially. In this manner, available computer resources, here processing time, are maintained at relatively high usage during all time periods, rather than experiencing

bursty usage patterns as conventionally occurs; hence, significantly enhancing overall system throughput. See Horvitz at col. 9, ll. 39-57

Claim 1 of the present application recites initiating examination of an instruction stream of a non-executing thread during execution of an executing thread, determining whether the hardware resource is available to the instruction of the non-executing thread and enabling execution of the non-executing thread if the hardware resource is available to the instruction of the non-executing thread. Applicant maintains that Horvitz does not disclose or suggest a process of *examining an instruction stream of a non-executing thread during execution of an executing thread and enabling execution of the non-executing thread if a hardware resource is available to the instruction of the non-executing thread.*

In fact, Horvitz discloses selecting a task instance at the beginning of idle-time intervals and processing them during the remainder of that interval. Thus, Horvitz teaches away from a process of *examining an instruction stream of a non-executing thread during execution of an executing thread*. The Examiner asserts that Horvitz does teach such a limitation at col. 11, ll. 24-26. The passage relied on by the Examiner recites:

> As noted above, **three distinct scenarios (characteristics) can arise for selecting task instances for precomputation: those future task instances that exhibit constant value over time, those future task instances that exhibit time-varying value, and those future task instances that exhibit higher present (expected) value than does a currently executing task instance. While the first two scenarios are directed to optimal use of idle-time and other processing periods of relatively low activity, the third scenario can arise and dictate optimal use of processing time during any period,**

> **regardless of whether it is low or high activity.** Though individually any task instance within a group of future task instances can exhibit any of these three value-based characteristics, for simplicity, I will separately address each of these characteristics and the decisional analysis used to select a future task instance from among a group of such instances having the same characteristic. Inasmuch as those skilled in the art will surely realize that the analysis, regardless of the modality used, reduces to a probability-based measure, **these different modalities, depending upon the characteristics of the tasks in a given group, can be combined as needed *during the onset of any idle-time interval to select an appropriate task instance,*** presently executing or future, from amongst those in the group, for current or continued execution, respectively.

Horvitz at col. 11, ll. 16-40.

The above-passage discloses a scenario where future task instances that exhibit higher present (expected) value than does a currently executing task instance dictates optimal use of processing time during any period, regardless of whether it is low or high activity. However, the passage also discloses that the future task instance is selected ***during the onset of any idle-time interval***. Moreover, if this scenario occurs, the future task instance is processed instead of the currently executing task instance, and ***not*** during execution of the currently executing task instance.

Additionally, Claim 1 of the present application recites identifying an instruction in an instruction stream, identifying hardware resources associated with the instruction and determining whether the hardware resource is available to the instruction of the non-executing thread. The Examiner maintains that Horvitz discloses such features at col. 17, ll. 52-67, which recites:

> Considering the net present value of results to be delivered by precomputing a future task instance necessitates time-discounting the net expected value of

> these results to the present. The discounted net expected value is then compared to losses in current value that would result from reducing or prematurely suspending the refinement of a current task instance. If the discounted value for any such future task instances) exceeds these losses, then a portion of present processing capacity is diverted from the presently executing task in favor of precomputing the future task instance(s)
>
> An immediate loss of dedicating current resources, such as processing time, over a period of time is the product of that resource and average EVC flux over that period. The gains of allocating these resources to a future task instance is a function of the total amount of idle-time that will be presently available after a current task is degraded . . .

Applicant submits that nowhere in the above-passage is there disclosed, or reasonably suggested, a process of identifying an individual instruction within a future task instance, or determining hardware resources available to that individual instruction. For the foregoing reasons, claim 1 is patentable over Horvitz.

Independent claims 13, 16 and 25 include features similar to those recited in claim 1. Thus claims 13, 16 and 25, and their respective dependent claims, are patentable over Horvitz for reasons similar to those recited in claim 1.

Claims 6 and 8 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Horvitz, in view of Budde et al. (U.S. Patent No. 4,891,753). Applicant submits that the present claims are patentable over Horvitz even in view of Budde.

Budde discloses register scorboarding on a microprocessor chip. Nonetheless, Budde does not disclose or suggest examining an instruction stream of a non-executing thread during execution of an executing thread, identifying an instruction in an instruction stream, identifying hardware resources associated with the instruction or determining whether the hardware resource is available to the instruction of the non-

executing thread. As discussed above, Horvitz does not disclose or suggest such features. Therefore, the combination of Horvitz and Budde would not disclose the feature. Accordingly, the present claims are patentable over Horvitz in view of Budde.

Claims 25, 26 and 27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Horvitz in view of Dukach et al. (U.S. Publication No. 2004/0036622). Applicant submits that the present claims are patentable over Horvitz even in view of Dukach.

Dukach discloses a system that shows messages on electronic displays, including networks of outdoor displays, such as displays mounted on vehicles. See Dukach at Abstract. However, Dukach does not disclose or suggest examining an instruction stream of a non-executing thread during execution of an executing thread, identifying an instruction in an instruction stream, identifying hardware resources associated with the instruction or determining whether the hardware resource is available to the instruction of the non-executing thread. As discussed above, Horvitz does not disclose or suggest such features. Therefore, the combination of Horvitz and Dukach would not disclose the feature. Accordingly, the present claims are patentable over Horvitz in view of Dukach.

Applicant respectfully submits that the rejections have been overcome, and that the claims are in condition for allowance. Accordingly, applicant respectfully requests the rejections be withdrawn and the claims be allowed.

The Examiner is requested to call the undersigned at (303) 740-1980 if there remains any issue with allowance of the case.

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: April 24, 2008

Mark L. Watson
Reg. No. 46,322

1279 Oakmead Parkway
Sunnyvale, California 94085-4040
(303) 740-1980